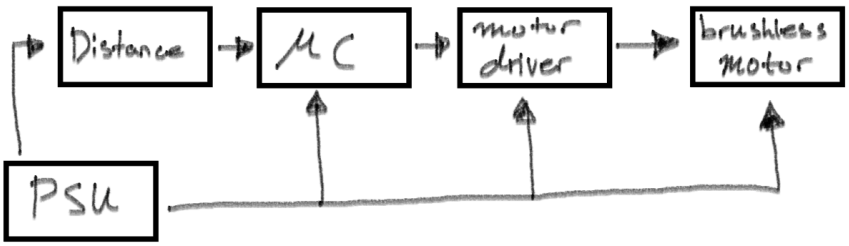
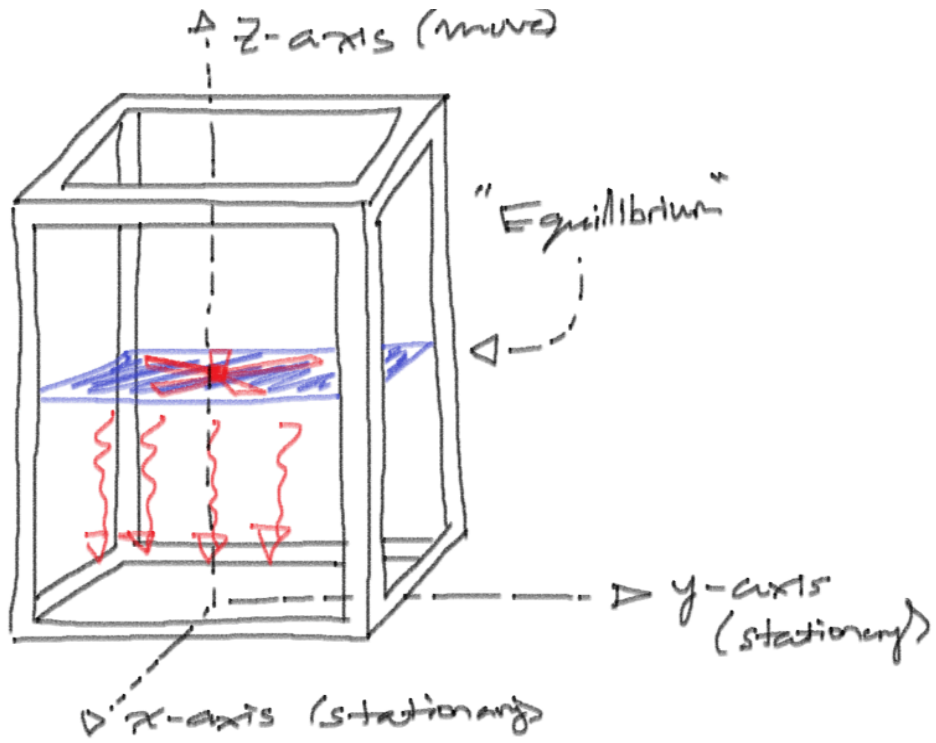
Logbook

09/09/2025

- The original idea was to make a drone, but it is way too complicated for one semester.
- Our project can be described as a dumbwaiter that uses a propeller instead of mechanical parts.
- We are using a single motor to power a propeller that goes up to a point considered "equilibrium". Weight, push, or pull forces can be done on the propeller. That would lead to the propeller increasing or decreasing speed to get back to the "equilibrium" point.
- The electronics we so far need are as follows: switches (x2), psu (x1), distance sensor (x1), brushless DC motor (x1), motor driver (x1), and microcontroller (x1).
    - Distance sensor we might use
    - Have to talk about it but might add a display
- A frame of sorts is going to be needed to keep the propeller stationary in the x and y direction, only moving in the z direction.
- Current block diagram:



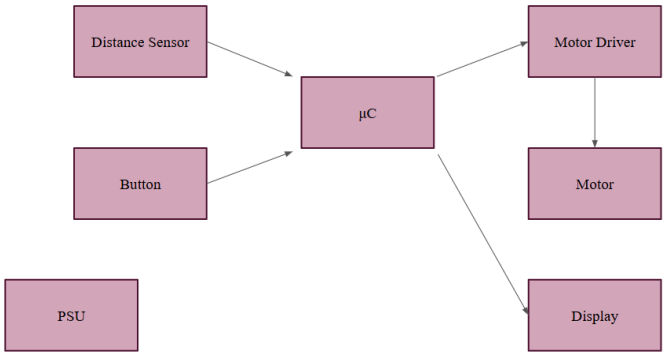
- Possible design:



Logbook

09/11/2025

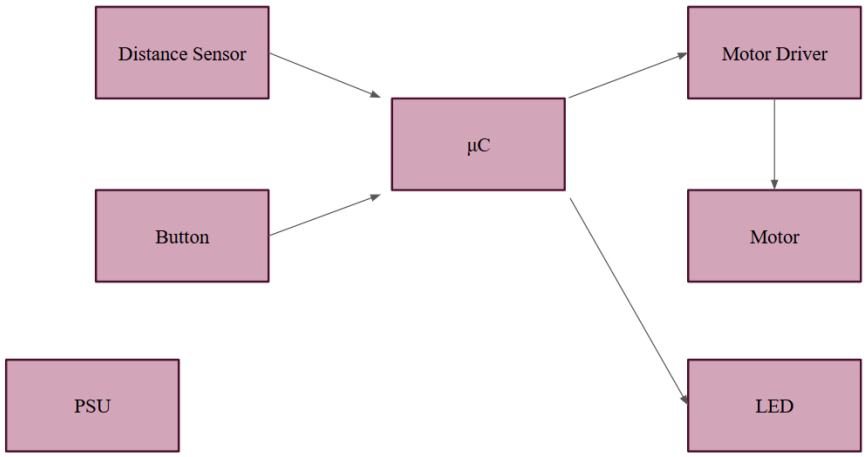
- There is another possible design for the frame that uses like 3 to 4 poles and that is what keeps the drone travelling only on the z-axis.
- We have decided that the drone will have all of its electrical components in one place which means the frame is just a structure to hold it in place.
- The button to turn the drone on and off will be on one side which keeps us away from using wires to connect all the electronics together.
- The electronics will probably mostly be placed in the center or in a way that the weight is evenly distributed.
- The motor and propeller we use will have to create enough downward force to move the drone up and down.
- The electronics we so far need are as follows: switches (x1), psu (x1), distance sensor (x1), brushless DC motor (x1), motor driver (x1), and microcontroller (x1). We also have to look into a voltage regulator since the drone would be running on a battery for power.
- Updated block diagram from last time:



Logbook

09/14/2025

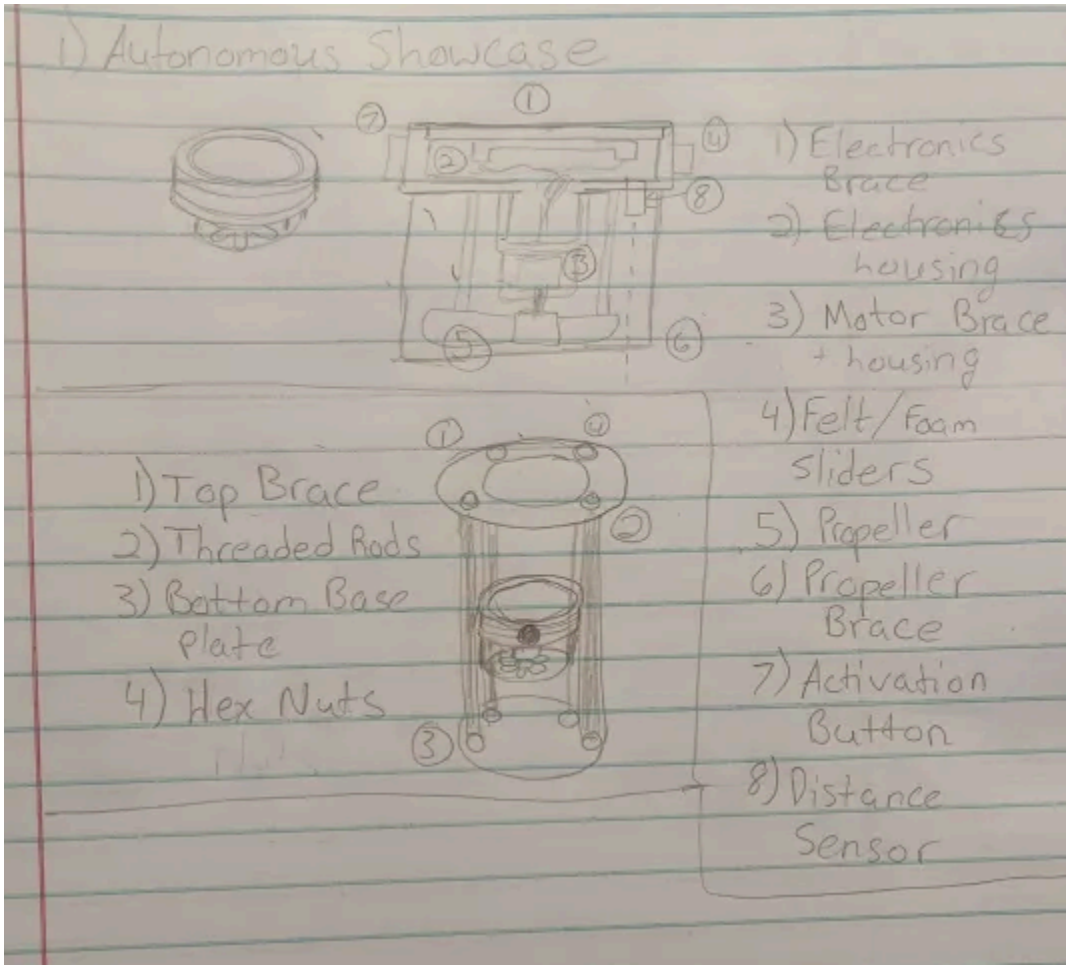
- New block diagram:



- Originally, we were thinking of using a display for giving user feedback about the drone but an LED was decided to be easier for this current prototype.
- The electronics we so far need are as follows: switches (x1), psu (x1), distance sensor (x1), brushless DC motor (x1), motor driver (x1), LED (x1), and microcontroller (x1). We also have to look into a voltage regulator since the drone would be running on a battery for power.
- Links to some electronics:
    - Microcontroller - Multiple ones we can choose from
    - Distance Sensor - The one professor talked about
    - LED - Any RGB LED will do, just have it there for reference
    - Button - Just have this button for reference, what ever is best
- Looking into motors, specifically for drones, there are some things to be aware of. Most specify the number of cells the LiPo battery has using “xS”. So 4S is 4 cells in series. To find the thrust they can produce, we can just look at the manufacturer’s info.
    - Possible motor with ESC - So the number of cells needed would be 4S and can max thrust 1024 g. From reading online, a good thrust-to-weight ratio is 2:1 for the basic movements of a drone which should be more than enough for us. The motor and ESC need like 4s which is almost 15 volts.
- A single LiPo cell is around 3.7 volts.
    - Possible battery
- Need a buck converter. If not, the battery will kill the ESP32.
- Possible names: AirLift or HMS (Height Maintenance System).

Logbook

- Other possible design:



Logbook

09/17/2025

- Concern arose about horizontal space dedicated to house the laser. Having the laser hang over the propeller would risk the distance data being skewed.
    - By pushing the laser to the outer perimeter of the propeller's range, we increase the overhang of the drone.
- Looked into the electronics we will use again and landed on this possible list:
    - Distance - https://www.adafruit.com/product/3317?srsltid=AfmBOophG-wvvBTgmJzAP-8jhGge4Poar271QmEji-KDwoUUaXPynfFA
    - Microcontroller - https://www.digikey.com/en/products/detail/espressif-systems/ESP32-S3-DEVKITC-1U-N8R8/16162636?gclsrc=aw.ds&gad_source=1&gad_campaignid=20243136172&gbraid=0AAAADrbLlgtgkG_Jm-L0YhXZI2s1SPaw&gclid=Cj0KCQjwuKnGBhD5ARIsAD19RsaObch_tm6VN9gXY12hmXCuD3Y-r59kiRYJ3PIVZ3_igxLuZIO8syMaApcsEALw_wcB
    - Motor - https://www.getfpv.com/brotherhobby-vy-1504-5-2650kv-2950kv-3950kv-motor.html
    - ESC - https://www.getfpv.com/v-good-rc-32-bit-30a-2-4s-brushless-esc-for-rc-airplane.html?afid=aVlOV0hBdmd6THc9&referring_service=google-cpc&utm_source=google&utm_medium=cpc&utm_campaign=DM%20-%20NB%20-%20PMax%20-%20Shop%20-%20No-index%20-%20SM%20-%20ALL%20%7C%20Full%20Funnel&utm_content=pmax_x&utm_keyword=&utm_matchtype=&campaign_id=20799936859&network=x&device=c&gc_id=20799936859&gad_source=1&gad_campaignid=20796067361&gbraid=0AAAAAD8cN5LjXoJDnIXa5zmCuLtj7zxSZ&gclid=Cj0KCQjwuKnGBhD5ARIsAD19RsYVdS8y1Dh9E5zQdFk1-bCXJuklTw65ExcxBI9Z00ZKhiOYvTiip78aAuW0EALw_wcB
    - Battery - https://www.getfpv.com/batteries/mini-quad-batteries/lumenier-550mah-4s-80c-lipo-battery-xt-30.html
    - LEDs and Buttons are just generic, don't have to be anything special
- Decided on two LEDs: one red and one green. This will tell the user when they can actually interact with the drone.

Logbook

09/18/2025

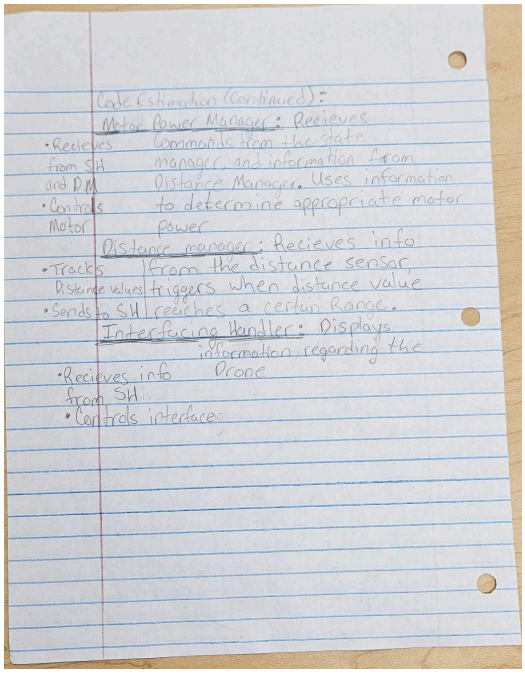
- Current Bill of Materials:

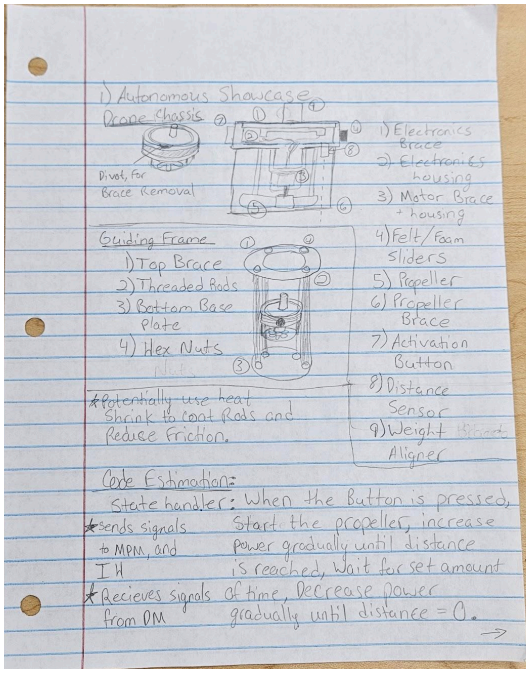
Electronic	***Link to Market Place***	***Status***	***Cost***
Microcontroller	ESP32 S3	Bought	
Motor	XING-E PRO 2207 2450KV	Available	
ESC		Available	
Distance	Adafruit VL53L0X Time of Flight Distance Sensor	Available	
PSU	Lumenier 550mAh 4s 80c Lipo Battery (XT-30)	Available	
Charger	Gens Ace iMars Mini G-Tech 60W 2-4S 5A	Available	
Propeller	Gemfan Hurricane SL 5125 3-Blade Propeller	Available	
LED	N/A	Bought	
Button	N/A	Bought	
Buck Regulator	N/A	Available	
LDO	N/A	Available	
3.5V Beeper	N/A	Available	
TVS diode	N/A	Available	
polyfuse at V in	N/A	Available	
capacitors	N/A	Available	
MOSFET	N/A	Available	

- Trying to buy most electronics to be able to begin getting things to run.
- Added extra components for the safety of the main electronics.
- https://esp32.com/viewtopic.php?t=20450 Info on how to interface with the ESC Using the ESP32. It might be a bit of a hassle, but according to this it's possible.

Logbook

09/24/2025

- Updated the README in github
- Buying the components by today and making sure they have datasheets to go with the electronics.
- Division of work for the project: Alec - Programming, Angelo - CAD, Cristian - PCB
- 

- 

- https://www.instructables.com/Build-Custom-ESP32-Boards-From-Scratch-the-Complet/
- Maybe it would be good to have two buttons: one that turns the drone on/off and another that initiates hovering and the other to land. The latter would have three states. The first is

standby, then hovering, and finally landing. After the landing, the button would go back to the standby state so the drone could be turned off. Just have to think of the code so the drone can turn on and off safely.

Logbook

09/30/2025

- Have two 4.7 k ohm resistors parallel to SDL and SDA lines connected to the voltage value of the ESP32. They can stay open assuming they don't end at some sensor.
- The GPIO pins for SDL and SDA for the ESP32 we are using are 21 and 22. 21 is SDA and 22 is SCL. The pins can be changed though.
- upesy.com/blogs/tutorials/esp32-pinout-reference-gpio-pins-ultimate-guide
- Have to check which GPIO pins are going to be used and which ones can be marked for no use.
- Waiting mostly on the ESC to be used to see if voltage regulators are needed or if the ESC has it itself.
- Distance Sensor Library
- ESP32 Servo Library

Logbook

10/01/2025

- Found some resources in order to make our own ESC for the actual prototype.
- We are going to have to use a ST-Link to program the STM chip with AM32. AM32 is the firmware used for STM chips which have a 32 bit ARM processor.
- https://www.youtube.com/watch?v=K9toWUsjgkE&list=PLoPtpxJIxgnbG5owAAyvgVvzenaRd1DPr&index=3
- https://www.youtube.com/watch?v=sau_KQx4EIA&list=PLoPtpxJIxgnbG5owAAyvgVvzenaRd1DPr&index=3
- https://www.youtube.com/watch?v=-ymTE-Nivzw&t=344s

Logbook

10/06/2025

- For the ESC, I decided to go with a different design from another guy because he shows actual testing.
- Changed the BOM and Design Decisions to reflect the electronics needed for the Proof of Concept. The extra electronics are for the actual PCB design and can be omitted for now.
- Since sensitive parts need 3.3V, and battery can run out and drop below that, a warning system for this is needed, hence a beeper starting at around 3.5V
    - planned part: 1-8S LiPo Low Voltage Buzzer Alarm
- Miscellaneous parts to foolproof it from getting fried from mistakes
    - 22-100 µF electrolytic on input to MP1584 buck, plus 0.1-1 µF ceramic at X5R/X7R
    - polyfuse at battery input to protect buck/ESC/etc. (to protect ESP from bursts of current)
    - p-channel MOSFET, backwards to block reverse polarity, as diode with low drop (to not fry stuff if battery is connected incorrectly)
- TVS diode is needed because plugging/unplugging LiPo, or motor switching noise, might be an issue due to voltage spikes
- Buck Regulator MP1584, sensitive parts (esp32, sensors) need 3.3V or they get fried
- LDO (one of the following parts will suffice; LDO needed to clean up noise of buck reg)
    - `AMS1117-3.3`
    - AP7333

Logbook

10/10/2025

- Changes to github adding design decisions and BOM documents for the POC. The POC code has also been added. The rough draft of the schematic is also added.
- The POC is mostly done and uses the items listed in the BOM.
- Got the motor to base its speed off the distance being read by the V53L0X. At the same time, there are 5 LEDs, with each one lighting up when the distance is increased by 200 mm. There is also a button used to turn on and off the motor and distance sensor without removing the power. Currently, the POC is connected to the USB port of my computer.
- After testing, the ESC is able to deliver enough power to make the POC run off the power of the battery and not be connected to a computer.
- Test that I did for the drone:
    - Test a blinking LED code
    - Test code for distance sensor
    - Test a blinking LED with button
    - Test distance sensor while turning LEDs on depending on the range of distance
    - Get the motor speed to change with the number of LEDs that are turned on depending on the range of distance being read by the distance sensor
    - Add a button to turn the device on or off
    - Use the battery as a source for the ESP32 as well

Logbook

10/21/2025

- Add ferrite beads before capacitors for noise reduction
- Put distance sensor in one of the corners away from the esc
- Shorten the distance between the esc, battery connector, and motor connector
- Could also try making the traces thicker, adding vias between two traces both carrying the battery voltage, also removing the mask on top of the traces so solder can be added to add wiggle room

Logbook

10/23/2025

- We are looking into doing tests for the code but would need to make a custom chassis. We are looking into making one out of legos (or similar toys) for an early design and then making two 3D prints. At the same time, making these early designs would make it easier for us to make the final prototype down the line, already knowing designs that work.
- The PCB design has been worked on in the background but still needs to make changes to further improve how it would fit into the chassis that is still being worked on and making sure the voltage and current is proper for all the electrical components.
- The weight of all the main electronics used in the POC is about 120g. These are going to be used to put it inside the custom chassis so code testing can begin.
- This is information about the screws: 1.5 mm size for the holes x 4.

Logbook

10/28/2025

- Change the schematic and PCB layout to correctly connect the crystal for the atmega chip
- Use two 18 pF capacitors parallel to each of the lines.
- ESC stopped working and now have to check if its soldering issues. Trying to fix it but also have a new one coming in case.
- To save space, the final design will only have pins (or sockets) to program the chips using FTDI. ESP32 has a sequence for the buttons in order to flash new information.
- For FTDI on the Atmega328p, five pins are going to be used and four for the ESP32.
- The Atmega328p can have a simple automatic circuitry for uploading new software to it but the ESP32 has to use the right combination of pressing the BOOT and RESET buttons.
- Need an external FTDI programmer and will use these pins: Rx, Fx, GRN, VCC, DTR. The last is only for the Atmega so manual buttons are not needed but are for ESP32
- Have the option of going with two MCUs, one for the ESC and another for talking between the sensors and ESC, or a single one that does everything
- Might have to switch to new ESC for testing the code

Logbook

12/11/2025

- The hovering works but the distance sensor is currently not on the plate so it's us physically changing the height.
    - Have to add the function to be able and land when another button is pressed.
- Have to make a new plate that holds the distance sensor on it as well and has longer wires.
- In the code, the PID values have to keep being changed to get satisfactory results.
- The frame made out of knex works good enough to show a presentation.
    - The height of the frame is 210 mm, so we can aim for like 180 mm of hovering height.
- All the electronics will be outside of the frame with wires being used to connect everything.

Logbook

12/17/2025

- The distance sensor and motor are on the same plate and the code works as intended.
- The frame sometimes gets stuck on the sides when trying to land but the hovering works as expected.
- Finishing the KiCAD and CAD models so they align with each other.